SCT HV

VME Model **HV203**

User Manual

Institute of Nuclear Physics (IFJ) declines all responsibility for damages or injuries caused by an improper use of the module due to negligence on behalf of the User.

It is strongly recommended to read thoroughly the User Manual before any kind of operation.

IFJ reserves the right to change partially or entirely the contents of this Manual at any time and without giving any notice

1 GENERAL INFORMATION 4

1.1 Introduction 4
1.1.1 Warranty 4
1.1.2 Contacts 4

1.2 General Description 4

1.3 Card Description 5
1.3.1 Card Controller 5
1.3.2 Card Addressing 5
1.3.3 Square wave generator. 5

1.4 Single Channel Description 5
1.4.1 Switch 5
1.4.2 Trafo 6
1.4.3 Rectifier, multiplier and filter 6
1.4.4 Regulator 6
1.4.5 Over-current protection 7
1.4.6 Voltage setting 7
1.4.7 Optocouplers 7
1.4.8 Voltage and current readout 7
1.4.9 Error amplifier 7

1.5 Specifications [3] 8

1.6 Front panel (Fig. 3) 8

1.7 References 8

2 INSTALLATION 10

2.1 Unpacking and Inspection 10

2.2 Safety 10

2.3 Hardware preparation 10
2.3.1 Handling procedure 10
2.3.2 Address setting 10
2.3.3 Module calibration 12

2.4 Operating Environment 13
2.4.1 Airflow and Cooling 13
2.4.2 Power Requirements 13

2.5 Cables 13

3 PROGRAMMING INFORMATION 14

3.1 Introduction 14

3.2 Channel microcontroller program description 14

3.3 Card microcontroller program description 14

3.4 User program of the crate controller 14
3.4.1 Functions 15

4 APPENDIX 1 17

1 GENERAL INFORMATION

1.1 Introduction

This manual describes the elements of the SCT High Voltage Power Supply – VME Model HV203, its operation and installation procedures.

1.1.1 Warranty

IFJ will repair or replace the module within the guarantee period one year from date of shipment if the Guarantor declares that the product is defective due to workmanship or materials and has not been caused by mishandling, negligence on behalf of the User, accident or any abnormal conditions or operations.

1.1.2 Contacts

In case of questions or problems contact the card designers:

Stefan Koperny, tel. +48 12 617 2996, E-mail: koperny@uci.agh.edu.pl

Edward.Gornicki@ifj.edu.pl
or
Piotr.Malecki@ifj.edu.pl

Tel.: +48 12 662 8010
Fax: +48 12 662 8458
Mail address: INSTITUTE of NUCLEAR PHYSICS
ATLAS EXPERIMENT LABORATORY
ul. Radzikowskiego 152
PL - 31-342 KRAKOW, POLAND

www.ifj.edu.pl/ATLAS/sct/scthv

1.2 General Description

The VME SCT High Voltage Power Supply is designed to provide the bias voltage for ATLAS SCT modules[1]. The HV203 model is a 1-unit wide, 6U height standard board. It houses 4 fully isolated channels, VME interface and square wave generator.
The system provides digitally controlled stable bias voltage in 0 - 500 V range and a precise measurement of the output current. A maximum load of each channel is 5 mA. A current trip limit can be set independently for every channel in the range from hundreds of nA to the maximum of 5 mA. Another parameter which can be selected individually for each channel is the ramping speed with which the nominal voltage is changed.
Ramping speed is digitally controlled and can be selected from a very wide range. Single channel functions are controlled by programmable microprocessor which communicates with a programmable card controller via fast serial link. Communication with the crate controller for the VME version uses standard addressing mode. Reaction times of processors to various conditions like requirement of a new setting, over-current and over-voltage trips etc. are well below 1 ms.

1.3 Card Description

One 6U card of the multichannel system of HV power supplies contains four identical channels. Some elements located on the card serve all channels: the card controller, the VME interface, square wave generator and circuits responsible for the distribution of power supplied by the power pack of the crate (Fig. 2).

1.3.1 Card Controller

The ATMEL flash microprocessor AT89C52 [5] is used as the card controller. It receives commands from the VME controller[2] and communicates with microprocessors of all channels located on the card. Communication with channels is realized via internal serial interfaces operating in the full duplex mode, asynchronously at 2 Mbaud rate. Transmission of bytes between card controller or channel controllers is buffered and serviced with the help of interrupts.

1.3.2 Card Addressing

For the full VME version the standard addressing mode is used. The card address corresponds to the eight VME address lines (A16 - A23) and can be set via the dip switch on each card. (see chapter 2.3.2)

1.3.3 Square wave generator.

Square wave generator is built on PWM (Pulse Width Modulator) chip SG3526 and two transistors IRF520. The chip's outputs OUTA (pin 13) and OUTB (pin 16) are set by potentiometers: **T** by P91, **t** by P90 as shown in Fig.1

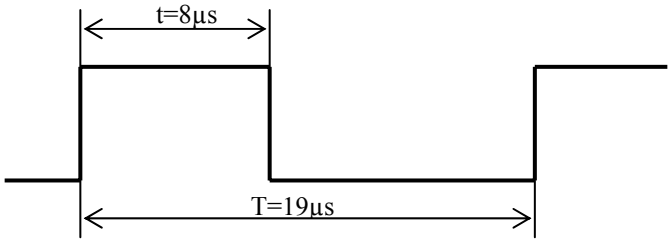

Figure 1. PWM outputs timing.

1.4 Single Channel Description

The single channel block diagram is shown in Fig. 2. It can be seen that the channel's full isolation is realized by optocouplers in the communication line and by transformer in the supply line.

1.4.1 Switch

The micro relay ZETTLER AZ850-5 (or equivalent)- controlled by the card processor is used to turn power on or off for the channel's transformer.

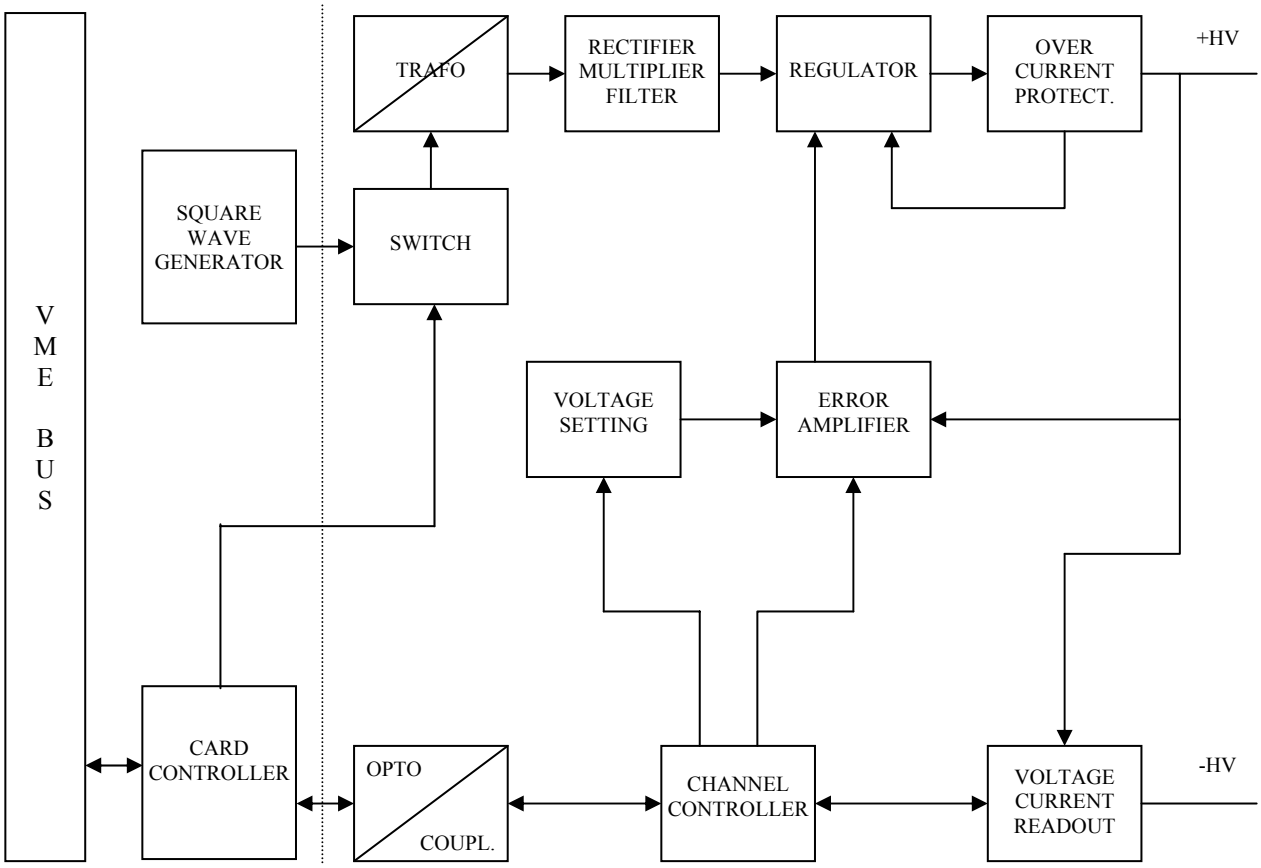

Figure 2 Block diagram of the HV channel

1.4.2 Trafo

The high frequency transformer (52 kHz, square wave) has one primary winding for U_{0m} of 58 V and three secondary windings of U_{1m} = 280V, U_{2m} = 12V and U_{3m} = 12V. The ferrite core EF16 is made of N67 material by Simens. U_{2m} and U_{3m} are auxiliary voltages supplying channel's circuitry.

1.4.3 Rectifier, multiplier and filter

A typical voltage multiplier (x2) is based on two high speed diodes UF4007 and two capacitors. The high voltage RC filter is applied for a ripple reduction.

1.4.4 Regulator

The high voltage MOSFET power transistor BUZ78 of U_{DSMAX} equal 800 V (catalogue value) is used as series regulator.

1.4.5 Over-current protection

An absolute (hardware) over-current protection is set to 5.2 mA. If the current exceeds this value the voltage is automatically reduced to keep the current below the limit. This protection is independent on the programmable current trip limit which is described below.

1.4.6 Voltage setting

Micro-controller can set the output voltage of the channel to the required value by setting the 10 bit serial DAC. This setting is always controlled via the ramp-up or ramp-down procedure. Ramp-up and ramp-down rates are to be chosen from several preset values. It may happen that the ramp-down rate will be determined by the time constant of the circuit rather than by the controller. Preset values are 5 V/s, 10 V/s, 20 V/s and 50 V/s.

1.4.7 Optocouplers

Two optocouplers 6N137 operating at 2 Mbaud inserted into the serial, asynchronous and duplex communication line provide electrical isolation between card and channel micro-controllers.

1.4.8 Voltage and current readout

The channel micro-controller continuously reads the output voltage and current using 12 bit multichannel ADC. The output voltage is measured with high precision voltage divider. Output current of the channel may vary from tens of nA to 5 mA. Every channel is equipped with the multi-range current readout system. A particular range is selected by the channel micro-controller to keep the readout accuracy high. These ranges are partially overlapping. One measurement cycle of voltage and current takes about 20 ms. The accuracy of the voltage measurements can be estimated from the following algorithm:
±(0.1% of the reading + 2 digits) and similar for the current accuracy: ±(1% of reading + 2 digits), where the significance of a digit is shown in column ``resolution'' in Table 1 for all ranges.

Table 1. Range and resolution of voltage and current measurement

	RANGE	**RESOLUTION**
Voltage	512V	1dgt=125 mV
Current	41.97 µA	1dgt=10.24 nA
	209.2 µA	1dgt=51.07 nA
	1.029 mA	1dgt=251.4 nA
	5.945 mA	1dgt=1.451 µA

1.4.9 Error amplifier

This block is based on the operational amplifier OP07 with a low input offset voltage. The voltage regulation loop can be opened or closed by the micro-controller. When the loop is open then the series regulator is cut off.

1.5 Specifications [3]

Table 2 Nominal Data

Parameter	*Value*	*Remark*
Output voltage	0 - 500 V	Programmable
Output current	0 - 5 mA	
Voltage ramping rates	50, 20, 10, 5 V/s	Programmable
Absolute current trip	5.2 mA	Hardware
Current trip limit	0 - 5 mA	Programmable
VME address mode	Standard (24 bits)	

1.6 Front panel (Fig. 3)

The green LED in the center of the front panel indicates that the card is powered from the VME crate. The reset button next to the green LED resets (i.e. restarts) the card controller.
Four fully isolated channels provide output voltage via LEMO connectors (ERA.0S.302.CLL) numbered from 0 - 3.
One red LED diode is located above each connector. The diode is off when the corresponding channel is not powered. After the channel is switched on but not set the LED blinks very slowly (twenty times a minute). The diode blinks also (several times a second) during the ramping up and about once a second during the ramping down of the channel voltage.
The LED stops blinking when the channel output voltage is stable.

1.7 References

[1] ATLAS Coll., Specification for Atlas Silicon Microstrip Detectors for the Atlas Final Design Review. ATLAS SCT/Detector FDR/99-2,1999
[2] The VME bus Handbook –Third Edition, Wade D. Peterson
[3] Atlas SCT HV Power Supply Project Specification, ver.2.04 http://www.ifj.edu.pl/ATLAS/sct/scthv/HVSpec_01Feb26.pdf
[4] ATMEL Data Sheet - 8-bit Microcontroller with 2k Bytes Flash AT89C2051
[5] ATMEL Data Sheet - 8-bit Microcontroller with 8k Bytes Flash AT89C52

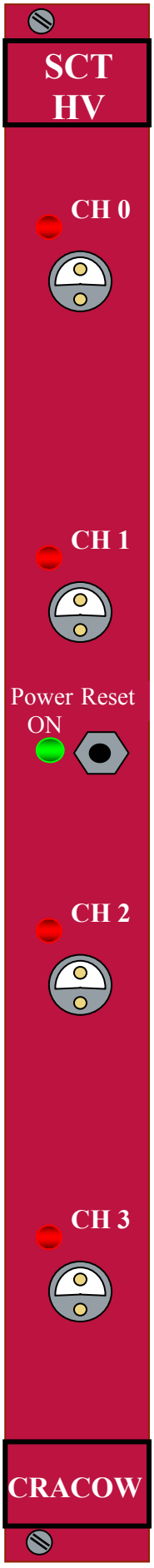

Figure 3. Front panel of the HV203 model

2 INSTALLATION

2.1 Unpacking and Inspection

Unpack module from shipping carton. If carton is damaged upon receipt, request that the carrier's agent be present during unpacking and inspection of the equipment. Refer to packing list and verify that all items are present. Save packing material for storing or reshipping the equipment.

2.2 Safety

Operating with HV203 board requires that everybody should keep the working rules for high voltage.
For the safety reason VME crate must be grounded.
Handling safety is obtained through careful design. There is no high voltage on the printed circuit board at solder side, HV components on component side are covered by heatsinks so that the user cannot accidentally be exposed to it so far, the board is placed into a VME crate.
All repairing must be done by qualified person.

2.3 Hardware preparation

The HV module should be inspected and prepared prior to system installation.
The following sections describe the proper address setting necessary for system operation.

2.3.1 Handling procedure

The HV203 board uses CMOS components that are susceptible to damage if exposed to static electrical charges. To avoid damage of these components during handling, testing or operation, the following procedure should be used.

- Device leads should contact conductive material to avoid building of any static charge, except during testing or operation.
- Soldering iron tips, metal fixtures, tools used in preparing the module for operation should be grounded
- Module should never be removed or inserted while power is applied to the module because voltage transients may permanently damage it.

2.3.2 Address setting

The module works in A24 mode; this means that the module address must be specified in a field of 24 bits.
The Address Modifiers recognised by the module are:

AM	=$39	:	Standard user data access
AM	=$3A	:	Standard user program access
AM	=$3D	:	Standard supervisor data access
AM	=$3E	:	Standard supervisor program access

Before inserting the VME HV203 board into a VME crate, the VME base address must be set.
The module's Base Address is fixed by DIP switch located on the board (see Fig. 4).

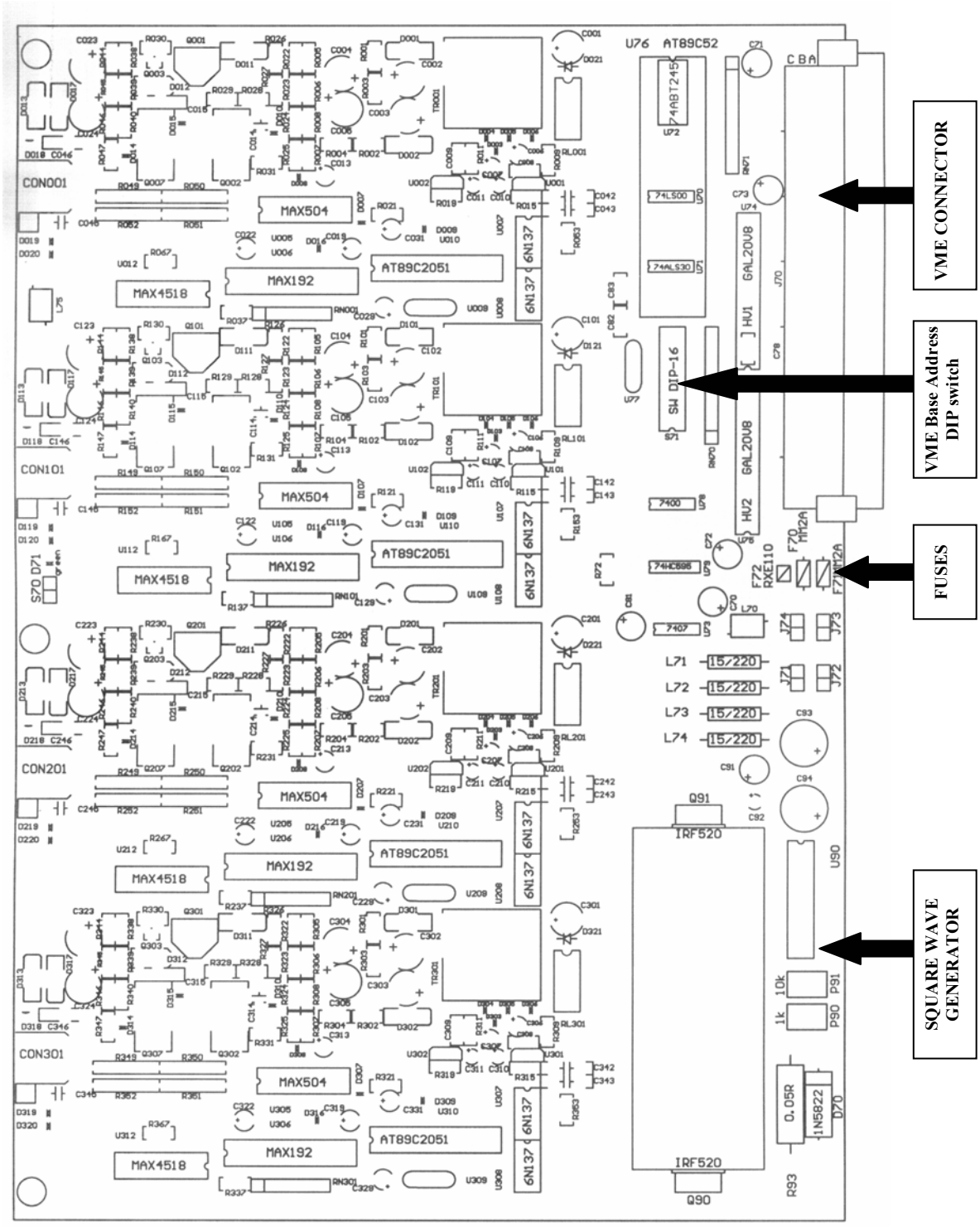

Figure 4. HV203 component side view

This is an eight_position switch which is used to control A16 - A23 VME address bits (see Fig. 5). Position numbered with 1 corresponds to the least significant bit of the corresponding part of the address. ON for a given position means that the corresponding bit is ZERO. So, e.g.: setting the positions 2 and 4 ON and all others OFF corresponds to the (part) of address equal to 0xF5.

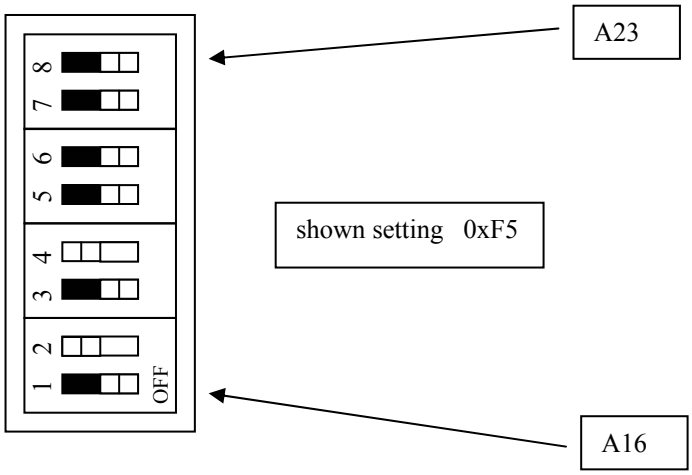

Figure 5. VME Base Address DIP Switch

2.3.3 Module calibration

The module is calibrated. ADC reference voltage correction is set by jumpers and DAC correction is made by software for each channel.
User calibration may be required in case of ADC (MAX192) replacement.
Follow this procedure to do it:

1. Set 500V for the channel.
2. Wait for about 30 min.
3. Test output voltage (use at least 4½ dig. voltmeter for 500V)
4. If the difference between digital output from computer and output from voltmeter is less then 0.1V the channel is correctly calibrated, if it is bigger go to step 5.
5. Two pads named J#05 (bottom side of the board) should be shorted if output voltage is lower then 500V or opened if higher then 500V.
6. Set appropriate number of bits to reach 500V, where 1bit corresponds to 125mV.
   Jumper positions form the binary value. When jumpers are set e.g. in positions #04 and #03 then it means that the correction value is equal to 12 bits. Correction equal to 5 bits require jumpers in positions #01 and #03.

2.4 Operating Environment

2.4.1 Airflow and Cooling

Standard VME crate airflow is required to maintain the operating temperature within specifications of the board. Failure to adhere to these requirements may result in poor long term reliability.

2.4.2 Power Requirements

+5V	300 mA
+12V	900 mA
-12V	900 mA

2.5 Cables

To connect any HV output to load, a cable one side ended by LEMO plug FFA.0S.302.CLAK.52 (CERN Store SCEM No: 09.31.30.060.4) must be used.
Particularly for ATLAS SCT application, for interconnection to SCT LV-2 (or SCT LV-3) module cable shown in Fig. 6 can be delivered on request (0.5m length default)

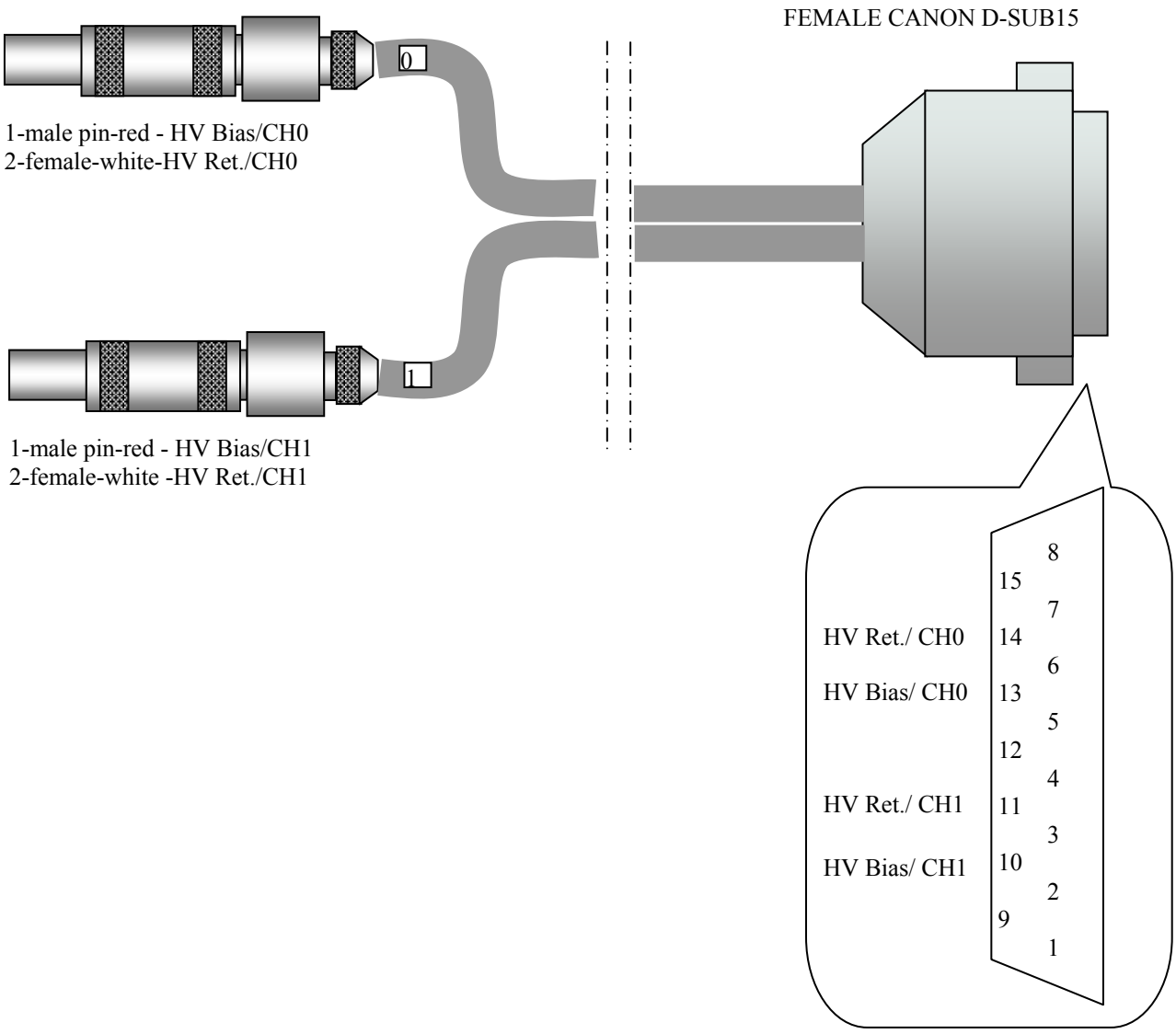

Figure 6. LV-HV interconnection cable

3 PROGRAMMING INFORMATION

3.1 Introduction

Operations of the VME SCTHV four-channel power supply card are controlled by software which is loaded into program memories of the channel microcontrollers, the card microcontroller and the VME crate controller. Two lower layers of this software system - programs of the channel and card controllers - are written in assembler for microcontollers of the family 51 and are loaded into flash memories of the ATMEL 89C2051 [4] and 89C52 [5] for channel and card controllers respectively. Card user does not communicate directly with these layers which can be treated as software emulated hardware functions.

3.2 Channel microcontroller program description

The program consists of the initialization part and of the endless loop part. The initialization part is executed once after power up or after card reset which causes the channel microcontroller reset. In this part various flags and initial conditions for timer, serial communication port and interrupt services are set. Five bits of the signed calibration constant for the ADC reference voltage, which are wired into the channel circuitry, are also read in and stored in this part of the program.

Main functions of the channel microcontroller program are realized in an endless loop in which the output voltage and current are permanently read from the corresponding channels of the 8-channel ADC. These permanent measurements include application of the proportional ADC correction and overcurrent and overvoltage tests. To keep the accuracy of the current measurements at required level over the very wide range of values program automatically selects also the optimal probe resistor.

This permanent loop is eventually interrupted by the card microcontroller requesting the current status and measurements or communicating new setting values. In case of new setting request the permanent loop is extended with the ramping control function.

3.3 Card microcontroller program description

The card controller program realizes the communication functions between the user program residing in the crate controller and channel microprocessors. Its organization is similar to the channel program. The part executed after power up or reset initialize timers, ports and various flags as well as clears the input buffer ready for new settings and four output data buffers, one for each channel.

The permanent loop is almost idle. Its main role is in some arbitration in case of conflicting transport requirements. Interrupts may come

- from the crate controller requesting current status and measurement data from a given channel,
- from the crate controller sending new settings for a channel,
- from the crate controller requesting switching ON or OFF the given channel
- from the channel controller confirming the reception of new settings
- from the channel controller sending the requested channel status and measurements,
- from the timer which takes care of frequent interrogation of active channels for the current data.

3.4 User program of the crate controller.

The crate controller program allows to handle and control our HVSCT power supply cards. It is usually written in C or C^{++} and depends in details on the operating system used to control the VME crate. It should provide the user with four principal functions to switch channel ON or OFF, to set the required voltage, ramping rate and over-current trip limit and to read the current channel data. A very simple example of a C programme used by designers for the VME crate with crate controller operating under OS9 system is given in the Appendix 1 and may serve as detailed information for programmers. As the SCT community uses NI VME/PCI interfaces and corresponding Peter W. Phillips programmes as certain standard we strongly recommend to observe his comments incorporated at the end of this Section.

3.4.1 Functions

The following description is important for preparing a program which allows to control the card. It is important that information passed from the user to the card controller via VME bus are prepared as described.
It is also important that the number of bytes transmitted at every operation is exactly as described.
Attached to this description (Appendix 1) is an example of a simple C program which was used by card designers.

Channel ON

Two bytes are send to the card controller. The first one is always 1 (means the number of bytes following). The second one contains a channel number (0 - 3) and command code.

Channel OFF

As above.

Setting the channel

The user specifies three values:
nominal voltage (in V)
current trip limit (in micro Amps)
ramping rate code (0: no ramping, 1: 50 V/s, 2: 20 V/s, 3: 10 V/s
4: 5 V/s).

This numbers have to be converted to binary representations according to the attached program. It is also important that the maxima are observed! *One should NEVER allow to set more then 500 V and 5000 micro Amps.* Observe also that over-current trip limits are recalculated for four different probe resistors, as in the attached program.
SET command requires that ALWAYS whole chain of 13 bytes is send to the bus (control byte, command/channel byte, 2 bytes of voltage, 8 bytes for the current limits and 1 byte of ramping rate code).
After the command is transmitted the card controller passes it to the corresponding channel and the channel starts ramping.

Observe, that the current trip limit has to take into an account the maximum current which may occur during the process of ramping as the test is executed permanently.

Reading the channel.

The channel processor reads the voltage and the current permanently.
It also takes care of automatic selection of one of four probe resistors to keep the measurement accuracy on a decent level. Status byte and results of measurements are reported by every active channel to the card controller on its frequent requests.

Reading the channel by the VME consists of

- sending two bytes (control byte and command/channel byte)
- receiving 10 bytes (control byte, status byte, two bytes of voltage readout , two bytes of current reading and four debug bytes for technical use).

Please, be sure that the proper number of bytes is written and read and that the binary to floating point numbers are converted as shown in the program example!

One also needs to allow for certain time delay between issuing the READ command and reading the buffer. My very rough estimation is that it should be a fraction of a milisecond.

Bits of the channel status byte have the following meanings:
bits 0 - 1 (least significant bits) contain the probe resistor number used for the current measurement,
bit 2 - reserve
bit 3 - reserve
bit 4 - unstable voltage
bit 5 - unrecognized command
bit 6 - tripped for the overcurrent
bit 7 - tripped for the overvoltage.

Comments by Peter W. Phillips

1. After switching on a channel, one must wait some time for the processor to reach its nominal state before sending any further commands. (Experimentally this seems to be a large fraction of a second after sending the "ON" command.)

2. NEVER send a ramp or status request to a channel which has not been switched on. I repeat, NEVER. This completely screws up the communication between the processors. (In retrospect, this should have been obvious....)

3. Regarding the time which must elapse between writing the two bytes to request the status of a given channel and reading back the 10 bytes of data, I find that this can be quite critical. Too short and it doesn't work, too long and it doesn't work. Between a rock and a hard place.
   Having deduced that the first of the ten data bytes represents the number of bytes which follow, I modified the code to run the following loop:

```
nloop=0;
nbyte=0;
while((nbyte!=9)&&(nloop<100)){
  //wait "rough fraction of mS"
  for(i=0;i<2000;i++){
    j++;
  }
  nbyte=*ip;
  nloop++;
}
```

4 APPENDIX 1

Header file : hvstd.h

```
#define VME_EXTENDED 0x00000000
#define VME_STANDARD 0xFF000000
#define VME_SHORT    0xFFFF0000
```

The program file

```
/* Simple communication programm in C for VME crate controller which reads
and writes data from/to HV_VME card */
#include    <stdio.h>
#include    <math.h>
#include    "hvstd.h"

main(argc,argv)
      int   argc; char  *argv[];
{
   unsigned char *addr, *wsk;
   double rv, fvolt, fcurr, voltdiv=1.25;
   double r10n[4];
   int vlow, vhigh, clow, chigh;
   int channel, voltage, current, maxcur, ramping;
   unsigned char status, ramp, vlows, vhighs, vlowr, vhighr;
   unsigned char chighr, clowr;
   unsigned char ctriph[4], ctripl[4];
   unsigned char mset, lset, mcorr, lcorr, nb;
   char c, znak, ii;
   int i,nbyte, lili, wype, karta;

/* probe resistors for the current measurements:       */
   r10n[0]=75050.0; r10n[1]=15050.0; r10n[2]=3050.0; r10n[3]=520.00;

   printf("\nENTER CARD ADDRESSS as 2 hex (e.g.: 08  A1) ");
   scanf ("%2x", &karta); karta=karta * 65536;

   for(;;){
      printf("\nn ON, f OFF, s SET, r READ; e.g: 2f means OFF the channel
2\n");
      scanf("%d %c", &channel, &c);
      switch(c){
                        /*  switch ON the channel    */

       case 'n':   if(channel<0 || channel>3 ) continue;
       ii=(unsigned char)channel;
       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D); *addr=0x01;
       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D);
*addr=16+channel;
      continue;
                        /*  switch OFF the channel    */

       case 'f':   if(channel<0 || channel>3 ) continue;
       ii=(unsigned char)channel;
       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D); *addr=0x01;
       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D);
*addr=channel;
      continue;
```

```
/* new settings for the channel: voltage, current_trip_limit and
ramping_rate */

       case 's':   if(channel<0 || channel>3 ) continue;
       printf("required voltage (nnn.n) in V "); scanf("%F", &rv);

/*-----do not change the sense of the following ----------------------*/

       if (rv < 0.0 ) rv = 0.0;
       if(rv > 500.0) {
          printf("max voltage: 500.0 V\n"); rv=500.0;
       }
       voltage=(int)(rv * 10.0/voltdiv);
       if (rv < 20.0) voltage=voltage+1;
       vhigh=voltage/256;  vhighs=(unsigned char) vhigh;
       vlow=voltage%256;   vlows= (unsigned char) vlow;

       printf("enter current_trip_limit in uA "); scanf("%d", &maxcur);
       if(maxcur < 0) maxcur = 0;
       if(maxcur > 5000){
          printf("max. prad: 5000 uA\n "); maxcur=5000;
       }
      for(i=0;i<4;i++){
/*  these are current_trip_limits (two bytes) for each probe resistor */
         current=(double)maxcur/1000.0 * r10n[i] * 1.3 ;
         chigh=current/256; clow=current%256;
         if(chigh > 15) {chigh=0x0F; clow=0xFF;
         }
/*       printf(" %02X%02X ", chigh, clow);   */
         ctriph[i]=(unsigned char) chigh;
         ctripl[i]=(unsigned char) clow;
      }
       printf("\nenter rumping rate code "); scanf("%d", &ramping);
/* 4 - slowest, i.e. 10V/s , 1 - fastest;  (secret: 0 - no ramping) */
       if((ramping <0)) ramping = 0;
       if((ramping >4)) ramping = 4;
       ramp=ramping;
       /*   write it down to VME bus  */
       addr=(unsigned char *)(VME_STANDARD + karta + 2*channel + 1);
*addr=0x0C;
       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D);
       *addr=32 + channel;
       *addr=vhighs;
       *addr=vlows;
       *addr=ctriph[0];
       *addr=ctripl[0];
       *addr=ctriph[1];
       *addr=ctripl[1];
       *addr=ctriph[2];
       *addr=ctripl[2];
       *addr=ctriph[3];
       *addr=ctripl[3];
       *addr=ramp;
       continue;

/*        READ the channel       status, voltage, current, some more   */

       case 'r':   if(channel<0 || channel>3 ) continue;
       addr=(unsigned char *)(VME_STANDARD + karta + 2*channel+1);
*addr=0x01;
       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D); *addr=48 +
channel;
```

```
       /* sorry for the following. I need some time before card controller
*/
       /* is ready with readdressing a buffer corresponding to the
requested channel*/
         for(lili=0;lili<200;lili++) wype=1;
       /* I guess that a fraction of a milisecond is enough (??)  */

       addr=(unsigned char *)(VME_STANDARD + karta + 0x000D);
       nb =*addr;
       status =*addr;
       vhighr =*addr;
       vlowr  =*addr;
       chighr =*addr;
       clowr  =*addr;
/* the following 4 bytes are for technical/debug purposes. One must read
them */
       mset  =*addr;
       lset  =*addr;
       mcorr =*addr;
       lcorr =*addr;
      printf("STATUS: %2x ", status);
       voltage=vhighr*256 + vlowr;
       fvolt=voltdiv * ((double) voltage )/10.0;
       current=chighr*256 + clowr;
       i=(int)(status & 0x03);
       fcurr=1000.00*((double)current)/r10n[i]/1.3;
       printf("\n voltage=%6.1f current=%6.2f\n", fvolt, fcurr);

       continue;


       default:    printf("\nwrong command\n");
         continue;
      };
   };
}
```